Discrete Applied Mathematics 181 (2015) 174–184

Contents lists available at ScienceDirect

Discrete Applied Mathematics

journal homepage: www.elsevier.com/locate/dam

Counting independent sets in a tolerance graph

Min-Sheng Lin *, Sheng-Huang Su

Department of Electrical Engineering, National Taipei University of Technology, 1, Sec. 3, Chung-Hsiao E. Rd., Taipei 106, Taiwan

ARTICLE INFO

Article history:
Received 22 April 2014
Received in revised form 9 September 2014
Accepted 10 September 2014
Available online 5 October 2014

Keywords:
Tolerance graphs
Independent sets
Maximal independent sets
Independent perfect dominating sets
Counting problem

ABSTRACT

Counting independent sets is a #*P*-complete problem for general graphs but solvable in polynomial time for interval and permutation graphs. This paper develops some polynomial time algorithms for counting independent sets, maximal independent sets, and independent perfect dominating sets in a tolerance graph, which is a common generalization of interval and permutation graphs. No algorithm for solving those problems for tolerance graphs is currently available.

1. Introduction

Let $G = (V, E)$ be a simple graph with set of vertices V and set of edges E. An *independent set* (abbr. IS) in a graph G is a subset W of V such that no two vertices of W are adjacent. The *maximal independent set* (abbr. MIS) in a graph is an IS that is not a subset of any other IS in the graph. A *dominating set* in a graph G is a subset D of V such that every vertex that is not in D is adjacent to at least one vertex in D. An *independent dominating set* in a graph G is a set of vertices of G that is both independent and dominating in G. Clearly, every dominating set that is independent must be maximal independent, so independent dominating sets are identically the maximal independent sets. An independent dominating set D is an *independent perfect dominating set* (abbr. IPDS) if every vertex that is not in D is adjacent to exactly one vertex in D.

This paper investigates the problems that are associated with the numbers of ISs in a graph. Provan and Ball [12] confirmed that counting ISs is #*P*-complete for general graphs and remains so even for bipartite graphs. Valiant [14] defined the class of #*P*-complete problems. The class of #*P* problems consists of problems that involve counting access computations for problems in *NP*, while the class of #*P*-complete problems includes the hardest problems in #*P*. As is widely known, all exact algorithms for solving these problems have exponential time complexity, and thus efficient exact algorithms for this class of problems are unlikely to exist. However, this complexity can be reduced by considering only a restricted subclass of #*P*-complete problems.

One very important special class of graphs is the class of intersection graphs. Let S be a finite family of non-empty sets. A graph G is an intersection graph for S if a one-to-one correspondence exists between the vertices of G and the sets of S such that two vertices are adjacent if and only if their corresponding sets in S have a non-empty intersection. The class of intersection graphs has various important subclasses. Some of them are briefly described below.

Chordal graphs are graphs in which every cycle with a length of at least four has a chord. Gavril [4] proved that chordal graphs are the intersection graphs of a family of subtrees in a clique tree. A tree T is a clique tree for a graph G if each node in T corresponds to a maximal clique in G and two nodes in T can be connected if corresponding maximal cliques intersect. For

* Corresponding author. Tel.: +886 953592110; fax: +886 229336508.
E-mail addresses: mslin@ee.ntut.edu.tw, mslin1234@gmail.com (M.-S. Lin).

http://dx.doi.org/10.1016/j.dam.2014.09.011

$v \in V$, let T_v be the set of all maximal cliques of G that contain vertex v. Therefore, G is a chordal graph if and only if T_v is a subtree in a clique tree T for every $v \in V$. In this way, four subclasses of chordal graphs can be defined [10]. *Undirected path graphs* are the intersection graphs of a family of undirected subpaths in a clique tree. *Directed path graphs* are the intersection graphs of a family of directed subpaths in a directed clique tree. *Rooted directed path graphs* are the intersection graphs of a family of directed subpaths in a rooted directed clique tree. A tree is called a rooted directed tree if one of its nodes has been designated as the root, and the edges are naturally orientated away from the root. Interval graphs are rooted directed path graphs in which the clique tree is itself a path. Interval graphs are usually defined as the intersection graphs of a family of intervals on a line. A graph $G = (V, E)$ is an *interval graph* if its vertices can be put into one-to-one correspondence with a set $\mathbb{I} = \{I_v | v \in V\}$ of closed intervals on a line such that two vertices are adjacent in G if and only if the corresponding intervals have a non-empty intersection; that is, $(u, v) \in E$ if and only if $|I_u \cap I_v| > 0$.

A *permutation graph* has an intersection model that consists of straight lines (one per vertex) between two parallel lines. *Trapezoid graphs* are the intersection graphs of a family of trapezoids (one per vertex) between two parallel lines. If every trapezoid is a line, then the intersection graph is a permutation graph. Similarly if every trapezoid is a rectangle, then the intersection graph is an interval graph. Thus, trapezoid graphs properly include both interval and permutation graphs. A *co-comparability graph* is the complement of a comparability graph. Co-comparability graphs are the intersection graphs of a family of curves (one per vertex) between two parallel lines. Corneil and Kamula [3] revealed that the class of trapezoid graphs is included in the class of co-comparability graphs.

A graph $G = (V, E)$ is a *tolerance graph* if there exist a set $\mathbb{I} = \{I_v | v \in V\}$ of closed intervals on a line and a set $\mathbb{T} = \{t_v | v \in V\}$ of positive real numbers, called tolerances, that satisfy the condition $(u, v) \in E$ if and only if $|I_u \cap I_v| \geq \min\{t_u, t_v\}$. The pair $\langle \mathbb{I}, \mathbb{T} \rangle$ is also called the *tolerance representation* of G. A vertex v in a tolerance graph is called a *bounded vertex*, if $t_v \leq |I_v|$. Otherwise, if $t_v > |I_v|$, v is called an *unbounded vertex*. A tolerance graph G is called a *bounded tolerance graph*, if all vertices of G are bounded. Notably, a tolerance graph with all $t_v = c$, where c is a fixed positive constant, is exactly an interval graph and a tolerance graph is exactly a permutation graph if all $t_v = |I_v|$ in its tolerance representation. Therefore, both interval graphs and permutation graphs are subclasses of bounded tolerance graphs. Every bounded tolerance graph is known to be a trapezoid graph [1]. However, general tolerance graphs are not included in trapezoid graphs.

Another generalization of interval graphs is the class of probe interval graphs. A graph is a *probe interval graph* if its vertices correspond to intervals, but every vertex is marked as either a probe or a non-probe. Two vertices in a probe graph are adjacent if their intervals overlap, and at least one of the vertices is a probe. Notably, every probe interval graph is a tolerance graph when infinite tolerances are assigned to non-probes and very small tolerances are assigned to probes.

Let n and m be the number of vertices and the number of edges in a graph respectively. Okamoto, Uno and Uehara [11] presented $O(n+m)$ time algorithms for counting ISs in a chordal graph, and also demonstrated that the problem of counting MISs remains #P-complete in a chordal graph. The present authors' recent work [8] showed that the problem of counting MISs remains #P-complete even when restricted to directed path graphs but a further restriction to rooted directed path graphs admits a solution in $O(n^3)$ time. Lin and Chen [7] presented $O(n^2)$ time algorithms for counting ISs and MISs in a trapezoid graph. Lin [6] proposed $O(n)$ time algorithms for counting ISs and MISs in an interval graph. The present authors' earlier work [13] derived $O(n^2)$ and $O(n^{2.3727})$ time algorithms for counting ISs and MISs, respectively, in a co-comparability graph.

Fig. 1 presents the containment relations among the aforementioned intersection graphs and summarizes the above results. However, the complexity of the problems of counting ISs and MISs remains unresolved for tolerance graphs and probe interval graphs. This paper is the first to demonstrate that these problems admit solutions in polynomial time for tolerance graphs. This paper also reveals that the problem of counting IPDSs is still solvable in polynomial time for tolerance graphs. Since the class of probe interval graphs is a subclass of tolerance graphs, all of these problems can be solved in polynomial time for probe interval graphs.

2. Preliminaries

This section presents the preliminaries on which the desired algorithms depend. Consider a tolerance $G = (V, E)$ with tolerance representation $\langle \mathbb{I}, \mathbb{T} \rangle$. For simplicity, let $V = \{1, 2, \ldots, n\}$ and let B and U be the sets of bounded and unbounded vertices in V, respectively. The following remarks are straightforward.

Remark 1. For $u \in U$ and $v \in B$, $(u, v) \in E$ if and only if $|I_u \cap I_v| \geq t_v$.

Remark 2. For $u, v \in U$, $(u, v) \notin E$.

Let $a(v)$ and $b(v)$ denote the left and right endpoints of interval I_v, respectively. Without loss of generality, the following assumptions are made. No two intervals share a common endpoint and set $t_u = \infty$ for any unbounded vertex u. The endpoints of all intervals are labeled with distinct positive integers. The vertices from 1 to n are labeled in a manner determined by their ascending right endpoints. That is, for two vertices i and j, $b(i) < b(j)$ if and only if $i < j$.

To simplify the implementation of the algorithm, two dummy bounded vertices 0 and $n+1$ are added to graph G, where $a(0) = b(0) = 0$ for vertex 0 and $a(n+1) = b(n+1) = \max\{b(v) | v \in V\} + 1$ for vertex $n+1$. Assume that dummy bounded vertices 0 and $n+1$ are isolated vertices, meaning that they are disconnected from all other vertices.